symmetry

MDPI

Article

Protecting the Intellectual Property of Speaker Recognition Model by Black-Box Watermarking in the Frequency Domain

Yumin Wang and Hanzhou Wu *

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; wym18721691@shu.edu.cn
* Correspondence: h.wu.phd@ieee.org

Abstract: Benefiting from the rapid development of computer hardware and big data, deep neural networks (DNNs) have been widely applied in commercial speaker recognition systems, achieving a kind of symmetry between "machine-learning-as-a-service" providers and consumers. However, this symmetry is threatened by attackers whose goal is to illegally steal and use the service. It is necessary to protect these DNN models from symmetry breaking, i.e., intellectual property (IP) infringement, which motivated the authors to present a black-box watermarking method for IP protection of the speaker recognition model in this paper. The proposed method enables verification of the ownership of the target marked model by querying the model with a set of carefully crafted trigger audio samples, without knowing the internal details of the model. To achieve this goal, the proposed method marks the host model by training it with normal audio samples and carefully crafted trigger audio samples. The trigger audio samples are constructed by adding a trigger signal in the frequency domain of normal audio samples, which enables the trigger audio samples to not only resist against malicious attack but also avoid introducing noticeable distortion. In order to not impair the performance of the speaker recognition model on its original task, a new label is assigned to all the trigger audio samples. The experimental results show that the proposed black-box DNN watermarking method can not only reliably protect the intellectual property of the speaker recognition model but also maintain the performance of the speaker recognition model on its original task, which verifies the superiority and maintains the symmetry between "machine-learning-as-a-service" providers and consumers.

Keywords: DNN watermarking; black-box; speaker recognition; artificial intelligence security

Citation: Wang, Y.; Wu, H. Protecting the Intellectual Property of Speaker Recognition Model by Black-Box Watermarking in the Frequency Domain. *Symmetry* **2022**, *14*, 619. https://doi.org/10.3390/sym14030619

Academic Editor: Jose Carlos R. Alcantud

Received: 28 February 2022
Accepted: 18 March 2022
Published: 20 March 2022

1. Introduction

In recent years, deep neural networks (DNNs) have achieved great success in many tasks, such as computer vision [1,2], speech recognition [3,4] and natural language processing [5], which prompts *machine learning as a service* to build a convenient bridge between service providers and service consumers. However, training a powerful DNN model is very expensive and requires: (1) a large amount of available data; (2) sufficient computing resources; and (3) experienced domain experts.

This implies that, as the key factor to achieve a kind of symmetry between "machine-learning-as-a-service" providers and consumers, a high-performance DNN model should be regarded as the intellectual property (IP) of the owner and be protected accordingly. Fortunately, increasing methods [6–12] have been proposed to protect DNN models by watermarking. Watermarking DNN models (also called *DNN watermarking*) is a technique that allows DNN model owners to embed ownership information in DNN models for IP protection.

A basic requirement is that the watermarking procedure should not impair the performance of the host DNN on its original task. Many DNN watermarking methods have been proposed along this line. DNN watermarking can be roughly divided into two categories, i.e., white-box DNN watermarking and black-box DNN watermarking.

White-box DNN watermarking [6–8] embeds the watermark information into the internal parameters or structures of the DNN model. As a result, in order to reconstruct the embedded watermark, the watermark extractor should be able to access the internal details of the target DNN model. A simple idea is to modify the network parameters for watermark embedding. However, simply modifying the network parameters is not desirable in practice as it may significantly impair the functionality of the DNN model. To this end, Uchida et al. [6] embedded watermark bits by optimizing a combined loss consisting of a task loss and a watermark loss.

During model training, by optimizing the combined loss, the watermark can be embedded into the network parameters while well maintaining the functionality of the DNN. A drawback is that the method imposes a statistical bias on certain model parameters, reducing the statistical imperceptibility of the watermark. Wang et al. [13] used those earlier-converged network parameters to carry a watermark, which can preserve the statistical characteristics better. Moreover, Wang et al. used an independent neural network for automatically watermark embedding and extraction, which skips complex manual operations and is more suitable for applications.

Different from the aforementioned methods that directly modify the network parameters by optimizing a loss, Rouhani et al. [7] proposed a watermarking method to embed secret bits into the probability density functions of different network layers, which shows better robustness against parametric attack. Wang et al. [8] proposed a watermarking method based on adversarial training. The watermarked model serves as the generator, whereas a watermark detector that detects changes in the statistical distribution of the model parameters serves as a discriminator.

During training, the generator is encouraged to generate non-detectable watermarks, whereas the detector tries to distinguish watermarked models from non-watermarked ones. Thus, the marked model tends to carry a watermark in a way that its parameter distribution stays similar to the non-marked one. Recently, Zhao et al. [14] proposed a method to embed watermark bits into the network structure. Compared with previous methods, this method can resist against all parameter-based attacks and thus has good potential in applications.

In contrast to white-box DNN watermarking, black-box DNN watermarking [9–12] enables verification of the ownership of the target DNN model without knowing the internal details of the model. It is often the case that the watermark extraction depends on the output results of the target model on a set of input samples. For example, Adi et al. [9] proposed a black-box DNN watermarking scheme by using backdoor technology. They use abstract images unrelated to the host DNN model as the trigger samples (also called backdoor samples) and train the host DNN model with normal samples and trigger samples.

As a result, the marked model not only performs very well on its original task but also enables the watermark to be verified by inputting the trigger samples. Clearly, there are different ways to construct the trigger samples, e.g., Zhang et al. [10] used text markers, noise samples and irrelevant samples as the trigger samples to verify the ownership.

The aforementioned methods can be regarded as *zero-bit watermarking* [15]. In order to realize *multi-bit watermarking*, Chen et al. [11] proposed a model-dependent encoding scheme that takes into account the owner's binary signature for watermarking. Guo et al. [12] proposed a watermarking method to protect the ownership of the DNN model deployed in embedded systems and designed a backdoor watermark with specific information generated by the bit array as a trigger.

Unlike the aforementioned methods that are originally designed for convolutional neural networks (CNNs), Zhao et al. [16] presented an efficient watermarking scheme for graph neural networks (GNNs), whose core is to train a GNN with a random graph as the trigger controlled by a secret key. In addition, apart from mainstream works that are focused on watermarking classification-based models, watermarking generative models have also been studied, e.g., [17]. For more black-box methods, we refer the reader to [15,18].

There is no doubt that mainstream methods have moved DNN watermarking ahead rapidly. However, most of these works are designed for image-related DNN models, and there is little study on speaker recognition model protection. As a research direction of speech signal processing, speaker recognition has been widely applied in many fields, such as speaker verification, judicial identification, speech retrieval, medical applications and military fields.

In particular, increasing speaker recognition models are based on DNNs. It is necessary to protect the IP of these DNN models. How to design a watermarking scheme for DNN based speaker recognition models is therefore crucial. Moreover, it is more desirable to design a black-box DNN watermarking scheme for speaker recognition models since in practical application scenarios, these models are easily stolen by attackers and packaged into application programming interfaces (APIs) for profits.

At this point, we can only interact by querying the APIs as the internal details of the speaker-recognition-related DNN model deployed in a commercial product is unavailable to us. Therefore, this motivates us to study black-box watermarking for speaker recognition models, which is more in line with a realistic application scenario.

In this paper, we designed a zero-bit black-box watermarking method. This method mainly uses the constructed trigger samples to query the target model and authenticate the ownership of the target model by comparing the predicted results with the pre-specified labels. This kind of method has been successfully applied in the field of computer vision [9,10]. Since the goal of this paper is to protect the IP of audio-based DNN, it is very difficult to directly migrate the existing method.

In addition, the trigger signals designed by many existing methods are perceptible, which are removable and easy to be attacked. When we use the trigger samples constructed on the basis of these trigger signals to query the API, it is easy to arouse the alert of the attacker and prevent us from querying, leading to the failure of model ownership authentication. To solve this problem, this paper proposes a construction method for an imperceptible trigger signal.

More specifically, there are two important problems to be addressed under the zero-bit black-box watermarking condition. One is how to craft the trigger samples, and the other is how to highly preserve the performance of the DNN model on the speaker recognition task. For the first problem, simply adding a noticeable pattern, such as predefined marker and meaningful content, will not only impair the imperceptibility of the watermark (which leads the watermark to be easily removed or attacked) but also gives the attacker a chance to forge fake trigger samples to confuse the ownership of the model.

To deal with this problem, in this paper, we carefully design the trigger audio samples in the frequency domain, which achieves good performance in both imperceptibility and robustness. For the second problem, we add a new label based on the existing labels and designate the trigger audio samples as the new label category to maintain the performance of the original task as unaffected as possible. The experimental results show that the proposed black-box DNN watermarking method can not only reliably protect the intellectual property of the speaker recognition model but also maintain the performance of the speaker recognition model on its original task well, which verifies the superiority and applicability of the proposed work.

The remainder of this paper is organized as follows. We first introduce the proposed work in Section 2. Then, we conduct convinced experiments and analysis in Section 3. Finally, we conclude this paper in Section 4.

2. Proposed Method

The proposed method includes three phases, i.e., trigger sample generation, watermark embedding and watermark verification. The goal of trigger sample generation is to design a universal method for constructing two sets of trigger audio samples that will be used for subsequent watermark embedding and watermark verification respectively. Notice that the two sets do not intersect with each other.

For watermark embedding, it is realized by training the speaker recognition model with a set of normal audio samples and a set of trigger audio samples. During training, the normal audio samples are associated with normal labels; however, all the trigger audio samples are associated with a new label that does not appear in the normal label set. After training, the resultant model is treated as *marked* and can be used. For watermark verification, by inputting a new set of trigger audio samples into the target model, we can verify the ownership by analyzing the outputted results of the target model. Figure 1 shows the general framework of the proposed method. In the following, we provide the technical details for each phase.

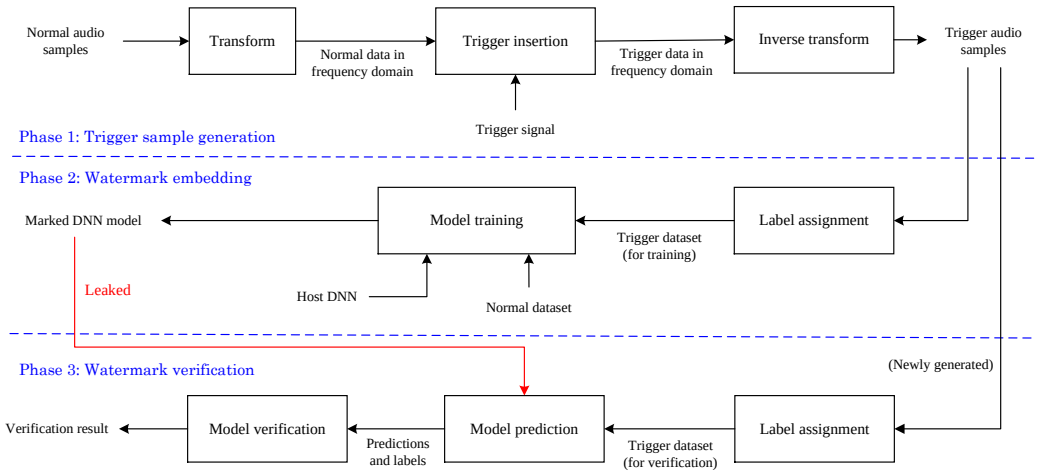

Figure 1. The general framework for the proposed method.

2.1. Trigger Sample Generation

Let $\mathcal{M}$ be the speaker recognition model to be marked. $\mathcal{M}$ accepts an audio sample $\mathbf{x} \in \mathbb{R}^L$ as the input and outputs the classification result $\mathcal{M}(\mathbf{x}) \in \mathcal{C} = \{0, 1, \ldots, c-1\}$, $c \geq 1$. In order to produce the marked model $\mathcal{M}^* \approx \mathcal{M}$, we need to construct a set of trigger audio samples. Most of the previous works construct trigger samples by adding a noticeable pattern (treated as a *trigger signal*) in the spatial domain of *normal samples* (also called *clean samples*), which has at least two drawbacks.

First, the noticeable pattern may arouse suspicion from the attacker and may allow the attacker to fake the trigger samples with the similar way. Second, once the attacker identifies the trigger pattern, he may attack the pattern, such as adding noise in the trigger sample to make the verification fail. In other words, adding a noticeable pattern in the spatial domain has low sample robustness. To deal with this problem, in this paper, we construct the trigger (audio) samples in the frequency domain, which was inspired by media watermarking in the frequency domain.

Conventional media watermarking can be roughly divided into two main categories, i.e., *spatial domain watermarking* and *transform domain watermarking* (typically also called *frequency domain watermarking*) [19,20]. Spatial domain watermarking refers to directly loading the watermark information onto the carrier data. For example, for speech signals, directly manipulating the amplitude value of the carrier speech time domain waveform at each moment, such as directly adding Gaussian noise to the carrier speech time domain waveform, is a spatial domain watermarking method.

Transform domain watermarking refers to loading the watermark information onto the coefficients of the transform domain, such as the Fourier transform domain or the wavelet transform domain of the carrier data. For example, for speech signals, the method of using the relevant knowledge of digital signal processing to perform discrete Fourier transform or discrete cosine transform on the carrier speech and then loading the watermark information

(such as text information or picture information) into the correlation coefficient obtained after frequency domain transformation belongs to the transform domain watermarking.

Compared with spatial domain speech (or audio) watermarking, frequency domain speech watermarking has several advantages [21]: (1) In the process of inverse transformation to obtain the marked content, the energy of the watermark information embedded in the frequency coefficient will be distributed to all moments in the time domain space, which has the effect of diluting the watermark information and is conducive to ensuring the invisibility of the watermark. (2) The human perception system is only sensitive to a few frequency bands; therefore, the careful manipulation of frequency domain coefficients can easily avoid the human auditory system from capturing changes or anomalies.

(3) The frequency domain method conforms to the international data compression standard; therefore, using the frequency domain method can easily implement the watermarking algorithm in the compressed domain and can resist the corresponding lossy compression. These advantages above indicate that designing frequency domain triggers may make the trigger signal not only imperceptible but also robust to attacks. Therefore, in this paper, we extend the frequency domain watermarking strategy to the construction of trigger audio samples.

In this paper, the idea of constructing the trigger audio samples is adding segment-based perturbation (corresponding to the trigger signal) in the frequency domain of normal audio samples. We produce a random sequence $\mathbf{t}$ containing 1, 0 and -1 with length l to be the trigger signal, which can be determined in advance, that is: $\mathbf{t} \in \{-1, 0, 1\}^l, l > 0$. In DNN based speech-related tasks, the input (audio) sample usually needs to be pre-processed by framing so that the data can be better processed later.

After framing, the subsequent training and testing operations on different frames of the input sample by the DNN model are independent of each other no matter the frames are carrying a trigger signal or not. We need to collect all the prediction results of the frames obtained from the DNN model for ownership verification. If we embed $\mathbf{t}$ in some segments of the frequency domain of the audio signal, the trigger signal will affect only a few frames of the audio signal but not all the frames; thus, the ownership verification may fail since the total number of frames not carrying the trigger signal may be significantly higher than the total number of frames carrying the trigger signal, which causes the verification result misled by the frames not carrying the trigger signal.

If we embed $\mathbf{t}$ once in some specific coefficients in the frequency domain, we may not guarantee that the feature pattern of the trigger signal is similar on different audio signals and can be learned by the DNN, which will also lead to the failure of watermark verification. To this end, we propose to embed $\mathbf{t}$ in each selected segment of the frequency domain of the audio signal so that the DNN model can reliably learn the mapping relationship between the trigger signal and the corresponding label during training, which enables the watermark to be extracted during verification.

Mathematically, given a normal audio sample $\mathbf{x} = \{x(0), x(1), \ldots, x(L-1)\} \in \mathcal{R}^L$, we first divide it into $R = \lceil L/l \rceil$ segments, if l does not divide L, the length of the last segment will be $L - (R-1) \cdot l$. At this point, $\mathbf{x}$ becomes $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_R\}$. Then, we process $\mathbf{x}_r$ $(1 \leq r \leq R)$ by discrete cosine transform (DCT) [22,23] to obtain the frequency domain coefficient $\mathbf{X}_r = \{X(0), X(2), \ldots, X(l-1)\} \in \mathcal{R}^l$, which can be expressed as:

$$\mathbf{X}_r(u) = \mathrm{c}(u) \sum_{i=0}^{l-1} \mathbf{x}_r(i) \cos\left[\frac{(i+0.5)\pi}{l} u\right], (0 \leq u < l), \tag{1}$$

where

$$\mathrm{c}(u) = \begin{cases} \sqrt{\frac{1}{l}} & u = 0, \\ \sqrt{\frac{2}{l}} & u \neq 0. \end{cases} \tag{2}$$

Then, we embed $\mathbf{t}$ into $\mathbf{X}_r$, which can be described as:

$$\mathbf{X}'_r = \mathbf{X}_r + \lambda \cdot \mathbf{t}. \tag{3}$$

where λ is a scalar used to control the intensity of the embedded $\mathbf{t}$. In this paper, λ is selected as 0.001, which is the optimal coefficient selected by comparing the experimental results after many experiments. During the experiment, in order to avoid artificially changing the DC component in $\mathbf{X}_r$, we specify $\mathbf{t}(0) = 0$. If l does not divide L, $L - (R-1) \cdot l$ bits of $\mathbf{t}$ will be embedded to $\mathbf{X}_R$.

Then, we perform inverse discrete cosine transform (IDCT) on $\mathbf{X}'_r$ to obtain the spatial trigger audio segment $\mathbf{x}'_r$ corresponding to the normal audio segment $\mathbf{x}_r$, i.e.,

$$\mathbf{x}'_r(i) = \sqrt{\frac{l}{2}} \mathrm{c}(i) \sum_{u=0}^{l-1} \mathbf{X}'_r(u) \cos\left[\frac{(i+0.5)\pi}{l} u\right], (0 \le i < l), \tag{4}$$

where

$$\mathrm{c}(i) = \begin{cases} \sqrt{\frac{1}{l}} & i = 0, \\ \sqrt{\frac{2}{l}} & i \neq 0. \end{cases} \tag{5}$$

Accordingly, we can concatenate all segments $\mathbf{x}'_r$ to construct $\mathbf{x}'$. The trigger audio sample will be used to train the DNN model. It is necessary to assign the trigger audio sample with a reasonable label. It is counter-intuitive and may degrade the performance of the original task if the category of the trigger audio sample is designated as any speaker other than the correct speaker in the existing speaker category set [24] since the timbre of the trigger audio sample is indistinguishable from the timbre of its corresponding original audio sample. In order not to affect the performance of the original task and to meet the requirements of convenient and reasonable watermark verification process, we assign a new label c to the trigger audio sample $\mathbf{x}'$. In other words, we extend the normal label set $\mathcal{C} = \{0, 1, 2, \ldots, c-1\}$ to a new label set $\mathcal{C}' = \{0, 1, 2, \ldots, c\}$.

2.2. Watermark Embedding

The goal of watermark embedding is to generate such a marked model $\mathcal{M}^* \approx \mathcal{M}$ that $\mathcal{M}^*$ not only has good performance on the original speaker recognition task but also remembers the mapping relationship between any trigger audio sample and the corresponding label. To realize this goal, we use a set of normal audio samples and a set of trigger audio samples to train $\mathcal{M}$ from scratch. Each of the normal audio samples is associated with the correct label.

However, each trigger audio sample is associated with the label "c" mentioned above. During model training, in each iteration, a mini-batch of random normal audio samples or trigger audio samples are fed to the model for each-round optimization. One thing to note is that, before training $\mathcal{M}$, a new label class corresponding to the trigger audio sample should be added to $\mathcal{M}$, which can be easily done by slightly modifying the softmax layer of $\mathcal{M}$. After training, the resultant model $\mathcal{M}^*$ is deemed *marked* and put into use.

2.3. Watermark Verification

As shown in Figure 1, by feeding a new set of trigger audio samples (together with the corresponding labels) to the target model, we are able to identify the ownership of the target model by comparing the prediction results and the assigned labels. Mathematically, with a total of n newly generated trigger audio samples $\{\mathbf{x}''_0, \mathbf{x}''_1, \ldots, \mathbf{x}''_{n-1}\}$, we determine their prediction results as $\{\mathcal{M}^*(\mathbf{x}''_0), \mathcal{M}^*(\mathbf{x}''_1), \ldots, \mathcal{M}^*(\mathbf{x}''_{n-1})\}$. It is noted that $\mathcal{M}^*(\mathbf{x}''_i) \in \mathcal{C}'$ for all $0 \le i < n$. Then, the ownership can be verified if

$$\sum_{i=0}^{n-1} \delta(\mathcal{M}^*(\mathbf{x}''_i), c) \ge \theta \cdot n, \tag{6}$$

where $\delta(x, y) = 1$ if $x = y$ otherwise $\delta(x, y) = 0$, and $\theta \in [0, 1]$ is a predetermined threshold close to 1. Otherwise, the ownership verification is deemed failed. It can be inferred that the verification phase does not require us to access the internal details of the target model, which can be, therefore, regarded as a black-box watermarking algorithm.

3. Experimental Results and Analysis

In this section, we provide experiments and analysis to evaluate the performance of the proposed method. We first introduce the experimental setup and then show the experimental results and analysis.

3.1. Experimental Setup

Our experiments are based on the popular TIMIT dataset [25], which uses the 3696 audio samples of 462 speakers. Among them, we used 80% of the samples for training and 20% for testing. Among the training samples, 10% of the samples were used as the validation set for selecting the best model. In addition, we constructed 36 new trigger audio samples constructed on the basis of 36 original audio samples corresponding to 16 randomly selected speakers from all eight American English dialects in the original dataset.

Among them, 16 trigger audio samples were added to the original dataset for watermark embedding, and these 16 trigger audio samples were added to the original training set and the original validation set in a ratio of 7:1. The 20 trigger audio samples were added to the original test set for watermark verification. Moreover, in order to ensure the validity of the experimental data, this paper preprocessed each audio sample, including the trigger sample, mainly removing the silent segments at the beginning and end of each sample.

It is free for us to choose the speaker recognition model to be marked. For the sake of simplicity, we used the well-known speaker recognition model SincNet [26] proposed by Mirco et al. for our experiments. The black-box watermarking framework proposed in this paper can be extended to other speaker recognition models and audio classification models. In these experiments, each audio sample was divided into frames with a frame length of 200 ms and a frame overlap of 10 ms, and the audio sample segment obtained after framing was used as the input of the model. In order to compare the experimental results in this paper with the experimental results in the literature, we adopted the same network structure and the same parameter initialization method as SincNet. The RMSprop optimizer [26] was used for parameter optimization with two hyper-parameters $\alpha = 0.95$ and $\epsilon = 10^{-7}$. The learning rate was set to $lr = 0.001$. The activation function used in all hidden layers was Leaky-ReLU [27], which not only achieves non-linear mapping but also avoids the possibility that some neurons will never be activated and the corresponding parameters will never be updated. Additionally, the batch size is 128, and the epoch size is 360.

For the speaker recognition task, we adopted two evaluation indexes, *frame-level error rate (FER)* and *sentence-level error rate (SER)*. Among them, the frame-level speaker classification results were obtained through the softmax layer, which provides the posterior probabilities of a set of speaker labels for each frame of the sentence. The result of sentence-level speaker classification is to use the voting method to select the prediction label with the maximum frequency among all the prediction labels corresponding to the frames as the final prediction label of the sentence. Regardless of the parameter θ shown in Equation (6), in general, we expect to keep both the FER and the SER as low as possible.

3.2. Results and Analysis

We evaluate the proposed method from three aspects: the performance on the original speaker recognition task, the performance on watermark verification and the ablation study.

3.2.1. Performance Evaluation on Speaker Recognition

In order to evaluate the performance of the proposed scheme on the original speaker recognition task, we compared the FER and SER obtained from the marked model $\mathcal{M}^*$ and the non-marked model $\mathcal{M}$. For these two models, we used the original test samples as input to obtain the FER and SER. The experimental results are shown in Table 1. By comparing the experimental results, it can be seen that the proposed method well maintained the speaker recognition performance (i.e., the speaker recognition accuracy was 99.3506% = 1 − 0.6494%), and there was little performance difference between $\mathcal{M}^*$ and $\mathcal{M}$.

Table 1. Performance comparison between the marked model $\mathcal{M}^*$ and the non-marked model $\mathcal{M}$ on the original speaker recognition task.

	Non-Marked Model $\mathcal{M}$	**Marked Model $\mathcal{M}^*$**
FER	47.9707%	48.8347%
SER	0.5772%	0.6494%

3.2.2. Performance Evaluation on Watermark Verification

In order to evaluate the performance of the proposed method on watermark verification, we fed a new set of trigger audio samples to $\mathcal{M}^*$ and observed whether the speaker category predicted by $\mathcal{M}^*$ was the assigned label. The experimental results are shown in Table 2. It can be seen that the proposed method had strong watermark verification ability (i.e., the success rate of watermark verification was 95% = 1 − 5%).

Table 2. Performance evaluation on the watermark verification task.

	A New Set of Trigger Audio Samples
FER	18.85%
SER	5.00%
Success Rate	95.00%

3.2.3. Robustness Evaluation on Watermark Verification

An adversary may attack the trigger audio samples used for watermark verification so that the copyright authentication cannot be carried out after these samples are input into the marked model, which requires evaluation of the robustness of the proposed method. To this end, we used noise insertion to mimic the real-world scenario. The basic Gaussian noise used here conforms to the standard normal distribution (i.e., $\mu = 0, \sigma^2 = 1$). Gaussian noise with different intensities can be obtained by specifying different signal–noise ratios (SNRs).

By adding Gaussian noise with different intensities to the trigger audio samples, we were able to determine the success rate of watermark verification, which can be used for evaluation. It is worth noting that we selected the trigger audio samples that can successfully verify the watermark in Section 3.2.2 for robustness analysis. The experimental results are shown in Table 3. It can be seen that the proposed work has good robustness (i.e., the success rate of watermark verification was 100%) even when the attack degree is strong.

Table 3. Robustness evaluation on the watermark verification task.

SNR(dB)	**SER**	**Success Rate**
5	0%	100%
10	0%	100%
15	0%	100%
20	0%	100%

4. Conclusions

In this paper, we proposed a black-box watermarking scheme for the speaker recognition model, which was able to not only complete the speaker recognition task but also protect the IP of the speaker recognition model. The proposed method produces two new sets of trigger audio samples. During the watermark embedding phase, the host model was trained from scratch with the combined normal audio samples and one set of trigger audio samples to embed the watermark. During the watermark verification phase, the other set of trigger audio samples was fed into the marked model to verify the watermark.

The experimental results demonstrated that the proposed work not only had a strong speaker recognition ability but also had a strong model ownership authentication ability, which achieved 99.3506% accuracy on the original speaker recognition task and achieved a 95% success rate on the watermark verification task, respectively. In addition, the experimental results also demonstrated that the proposed work could resist noise attacks to a certain extent, which verified the superiority and applicability. We hope that this work will contribute to the IP protection of speaker recognition models and lead to more advanced work in the future.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W. and H.W.; software, Y.W. and H.W.; validation, H.W.; supervision, H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) under Grant No. 61902235 and the Shanghai "Chen Guang" Project under Grant No. 19CG46.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
3. Xiong, W.; Wu, L.; Alleva, F.; Droppo, J.; Huang, X.; Stolcke, A. The Microsoft 2017 conversational speech recognition system. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5934–5938.
4. Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; Bengio, Y. End-to-end attention-based large vocabulary speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4945–4949.
5. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
6. Uchida, Y.; Nagai, Y.; Sakazawa, S.; Satoh, S. Embedding watermarks into deep neural networks. In Proceedings of the ICMR'17: International Conference on Multimedia Retrieval, Bucharest, Romania, 6–9 June 2017; pp. 269–277.
7. Rouhani, B.D.; Chen, H.; Koushanfar, F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural ne works. In Proceedings of the ASPLOS'19: Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, 13–17 April 2019; pp. 485–497.
8. Wang, T.; Kerschbaum, F. Robust and undetectable white-box watermarks for deep neural networks. *arXiv* **2019**, arXiv:1910.14268.
9. Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 1615–1631.
10. Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M.P.; Huang, H.; Molloy, I. Protecting intellectual property of deep neural networks with watermarking. In Proceedings of the ASIA CCS'18: ACM Asia Conference on Computer and Communications Security, Incheon, Korea, 4 June 2018; pp. 159–172.
11. Chen, H.; Rouhani, B.D.; Koushanfar, F. Blackmarks: Blackbox multibit watermarking for deep neural networks. *arXiv* **2019**, arXiv:1904.00344.

12. Guo, J.; Potkonjak, M. Watermarking deep neural networks for embedded systems. In Proceedings of the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, USA, 5–8 November 2018; pp. 1–8.
13. Wang, J.; Wu, H.; Zhang, X.; Yao, Y. Watermarking in deep neural networks via error back-propagation. In: *Electronic Imaging, Media Watermarking, Security and Forensics*; Society for Imaging Science and Technology: Springfield, VA, USA, 2020; pp. 22-1–22-9.
14. Zhao, X.; Yao, Y.; Wu, H.; Zhang, X. Structural watermarking to deep neural networks via network channel pruning. In Proceedings of the 2021 IEEE International Workshop on Information Forensics and Security (WIFS), Montpellier, France, 7–10 December 2021; pp. 1–6.
15. Li, Y.; Wang, H.; Barni, M. A survey of deep neural network watermarking techniques. *arXiv* **2021**, arXiv:2103.09274.
16. Zhao, X.; Wu, H.; Zhang, X. Watermarking graph neural networks by random graphs. In Proceedings of the 2021 9th International Symposium on Digital Forensics and Security (ISDFS), Elazig, Turkey, 28–29 June 2021; pp. 1–6.
17. Wu, H.; Liu, G.; Yao, Y.; Zhang, X. Watermarking neural networks with watermarked images. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2591–2601. [CrossRef]
18. Boenisch, F. A survey on model watermarking neural networks. *arXiv* **2020**, arXiv:2009.12153.
19. Dabas, P.; Khanna, K. A study on spatial and transform domain watermarking techniques. *Int. J. Comput. Appl.* **2013**, *71*, 2013. [CrossRef]
20. Kong, Y.; Zhang, J. Adversarial audio: A new information hiding method and backdoor for dnn-based speech recognition models. *arXiv* **2019**, arXiv:1904.03829.
21. Li, M.; Zhong, Q.; Zhang, L.Y.; Du, Y.; Zhang, J.; Xiang, Y. Protecting the intellectual property of deep neural networks with watermarking: The frequency domain approach. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December–1 January 2021; pp. 402–409.
22. Zong, Q.; Guo, W. A speech information hiding algorithm based on the energy difference between the frequency band. In Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, 21–23 April 2012; pp. 3078–3081.
23. Jeyhoon, M.; Asgari, M.; Ehsan, L.; Jalilzadeh, S.Z. Blind audio watermarking algorithm based on DCT, linear regression and standard deviation. *Multimed. Tools Appl.* **2017**, *76*, 3343–3359. [CrossRef]
24. Zhong, Q.; Zhang, L.Y.; Zhang, J.; Gao, L.; Xiang, Y. Protecting IP of Deep Neural Networks with Watermarking: A New Label Helps. *Adv. Knowl. Discov. Data Min.* **2020**, *12085*, 462–474.
25. Garofolo, J.S.; Lamel, L.F.; Fisher, W.M.; Fiscus, J.G.; Pallett, D.S.; Dahlgren, N.L. DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM. 1993. Available online: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir4930.pdf (accessed on 10 January 2022).
26. Ravanelli, M.; Bengio, Y. Speaker recognition from raw waveform with sincnet. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 1021–1028.
27. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 3–8.